#2

PATENT
Docket No. DE9-2000-0031 (267)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of FISCHER, et al.

Application No.		Examiner:
Filed:	(Herewith)	Group Art Unit:
For:	METHOD AND SYSTEM FOR THE AUTOMATIC GENERATION OF MULTI-LINGUAL SYNCHRONIZED SUB-TITLES FOR AUDIOVISUAL DATA	

CLAIM OF FOREIGN PRIORITY

Box Patent Application
Assistant Commissioner for Patents
P.O. Box 2327
Arlington, VA 22202-3513

Sir:

Priority under the International Convention for the Protection of Industrial Property and under 35 U.S.C. §119 is hereby claimed for the above-identified patent application, based upon European Application No. 00126893.7 filed December 7, 2000, and a certified copy of this application is submitted herewith which perfects the Claim of Foreign Priority.

Respectfully submitted,

Date: 11/27/01

Kevin T. Cuenot
Gregory A. Nelson, Registration No. 30,577
Kevin T. Cuenot, Registration No. 46,283
Steven M. Greenberg, Registration No. 44,725
AKERMAN SENTERFITT
222 Lakeview Avenue
Post Office Box 3188
West Palm Beach, FL 33402-3188
Telephone: (561) 653-5000

Docket No. DE9-2000-0031 (267)

Express Mailing Label No. EL 740159514 US

Europäisches Patentamt

European Patent Office

Office européen des brevets

Bescheinigung

Certificate

Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. **Patent application No.** **Demande de brevet n°**

00126893.7

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

Hatten

I.L.C. HATTEN-HECKMAN

DEN HAAG, DEN
THE HAGUE, 31/05/01
LA HAYE, LE

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.:
Application no.: 00126893.7
Demande n°:

Anmeldetag:
Date of filing: 07/12/00
Date de dépôt:

Anmelder:
Applicant(s):
Demandeur(s):

International Business Machines Corporation
Armonk, NY 10504
UNITED STATES OF AMERICA

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:

Method and system for the automatic generation of multi-lingual synchronized sub-titles for audiovisual data

In Anspruch genommene Prioriät(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat: State: Pays:	Tag: Date: Date:	Aktenzeichen: File no. Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:

/

Am Anmeldetag benannte Vertragstaaten:
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE/TR
Etats contractants désignés lors du depôt:

Bemerkungen:
Remarks:
Remarques:

DESCRIPTION

Method and System for the Automatic Generation of Multi-Lingual Synchronized Sub-Titles for Audiovisual Data

BACKGROUND OF THE INVENTION

The present invention relates to the field of multimedia data handling and, more specifically, to a computer-based method and system for performing synchronized sub-titling of audio or audio-visual material in a multitude of different languages.

Most of todays moving picture productions like movies, commercials or the like aim at an international audience and thus are not only distributed in the language used during production. Moreover additional synchronized language versions are assembled in which all dialogues and narrative sequences are spoken in the mother tongue of a particular audience.

Since the time efforts and costs for producing such synchronized versions are substantial, these language adaptations are usually produced only for the world's major languages. But for the languages spoken in smaller market segments either the original production or a sub-titled version is used. In a sub-titled version the translation of the spoken components appears as a synchronized textual overlay, usually in the lower area of the image. Hereby the term 'sub-title' is understood to refer generally to a textual, time-tagged representation that has to be rendered to become visible or audible.

Thereupon, modern distribution media like the DVD allow to assemble an international version of any audio-visual material that contains both, multiple sound-tracks for different languages as well as multiple sub-title streams that can be selected and switched by the user during playback. Currently most of this information has to be generated and synchronized manually.

SUMMARY OF THE INVENTION

It is therefore an object of the present invention to provide a computer-assisted or computer-based method and system for synchronizing a realization of a media stream having a synchronized first representation with at least one second representation.

It is another object to provide such a method and system which enable to perform the synchronization of the realization with the at least one second representation automatically.

It is yet another object to provide such a method and system for generating further synchronized sub-titles for media streams with an existing first sub-title.

The objects are solved by the features of the independent claims. Advantageous embodiments of the invention are subject matter of the dependent claims.

The present invention accomplishes the foregoing by building time-synchronous links between the realisation of the media stream, i.e. the audio or video material itself, and the representations, for instance the textual representations of the words spoken in the audio-visual material. The idea underlying the invention hereby is to provide a synchronization between the realisation and a first representation and to inherit the synchronization information to an at least second representation of the realization wherein using structure association information determined between the first and the at least second representation. Hereby it is emphasized that the term inheritance is understood in the broadest sense and by no means limited to the same term used in modern programming languages.

The invention thus allows for an automatic computer-based or computer-assisted generation of sub-titled or synchronized versions of audio-visual material in languages other than the one used during production. It is noteworthy that the sub-titles generated by use of the proposed mechanism may be rendered not only by overlaying the video with a written representation of the spoken data but may be fed into a system generating a rendering in sign language for the hearing impaired (in some countries such a version is a legal requirement for public broadcast) or even fed into a speech-synthesis system to become audible.

It should further be noted that the automatic generation and synchronization of language adapted versions is by no means restricted to the prementioned movie industry. For instance, in language-learning applications, the above mentioned technique can be used to synchronize an inter-linear translation with the audio material in order to better facilitate a student's understanding of the material.

Thereupon, in the field of Digital Audio Broadcasting (DAB), the proposed method and system allow to transmit the original version of e.g. an interview together with a translation of it as running text visible in the display of a DAB receiver. In addition, an ever increasing group of companies use TV broadcasting for an internal business TV. At least those companies operating on an international scale have to cope with the need of providing translations or sub-titles in a plenitude of languages.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention will be understood more readily from the following detailed description when taken in conjunction with the accompanying drawings, in which:

Fig. 1 is a schematic block diagram of an aligner known in the prior art that can be used for the present invention;

Fig. 2 is a flow chart diagram illustrating a method for combining a representation and a realization according to the prior art;

Fig. 3 is a high-level flow diagram illustrating the basic concept of the present invention;

Fig. 4 is a block diagram depicting a high-level structure of a system for generating subtitles according to a first embodiment of the invention;

Fig. 5 is a block diagram similar as in Fig. 4 depicting a high-level structure of the system according to a second embodiment of the invention;

Fig. 6a,b are block diagrams depicting a more detailed structure of the first embodiment of the system according to the invention;

Fig. 7 shows a document structure of an audio as generated by performing speech/non-speech segmentation, speaker segmentation, and phrase segmentation;

Fig. 8 shows a typical output of an audio structurer detecting speech/non-speech transitions, speaker changes, and phrases (Segment labels: M mixed, N non-speech, S speech, n speaker n, P phrase);

Fig. 9 shows a document structure of an audio as determined by the audio structurer of the first embodiment where Nodes are labeled by segment type (M: mixed, N: non-speech, S: speech) and store information about the location of the segment boundaries (subscript s);

Fig. 10 is a flow diagram of a text structurer for plain text;

Fig. 11 is a flow diagram of a text structure aligner according to the invention;

Fig. 12 is a text sample (Donna Leon, Acqua alta, Macmillan 1996) and it's translation (Donna Leon, Acqua alta, Translation: Monika Elwenspoek, Diogenes Verlag 1997);

Fig. 13 is a structured text sample (Donna Leon, Acqua alta, Macmillan 1996) and its translation (Donna Leon, Acqua alta, Translation: Monika Elwenspoek, Diogenes Verlag 1997);

Fig. 14 is an exemplary mapping table between a Tree Locator and Tree Locator ID;

Fig. 15 is an exemplary mapping table between a Tree Locator and Text;

Fig. 16 shows a document structure of the text sample (Fig. 12) as determined by the text structurer of the first embodiment of the invention where sentence nodes are labeled by the number of dependent word nodes;

Fig. 17 shows a document structure of the translation of the text sample as determined by the text structurer of the first embodiment where sentence nodes are labeled by the number of dependent word nodes;

Fig. 18 shows an alignment of the document structures for text and translation for the above text sample expressed as HyTime links.

Fig. 19 is a flow diagram depicting a link generator according to the invention;

Fig. 20 shows exemplary links for the alignment of text, translation, and audio of the sample text (Fig. 12) expressed in SGML syntax;

Fig. 21 is a flow diagram depicting an exemplary algorithm performed by a renderer for generating a view from a link-web;

Fig. 22 is an example of a synchronisation expressed by HyTime links in accordance with standard ISO 10744;

Fig. 23 is an exemplary output of the step of analyzing the structure of two representations A and B and synchronizing the two revealed structures with each other in order to build structure links between equivalent structural elements of both representations A and B; and

Fig. 24 exemplary links based on which a synchronization can be used to provide a time-synchronous alignment between a representation B and a video/audio stream.

DETAILED DESCRIPTION OF THE DRAWINGS

A mechanism to generate a synchronization between a text and corresponding audio data is disclosed in U.S. Patent Application No. ____________ (docketno. DE9-1999-0053) which is fully incorporated herein by reference.

Therein disclosed mechanism particularly allows for creating links (hyper links) between a representation, e.g. the text data, and a realization, e.g. the corresponding audio data. Hereby the realization is structured by combining a time-stamped version of the representation generated from the realization with structural information from the representation.

The known mechanism is based on the fact that most acoustic multimedia data have a common property which distinguishes them from visual data. These data can be expressed in two equivalent forms: as a textual or symbolic representation, e.g. score, script or book, and as realizations, e.g. an audio stream. The realization therefore can be structured by combining a time-stamped (or otherwise marked) version of the representation generated from the realization with structural information from the representation. Errors within the time stamped representation are eliminated by aligning the time-stamped version of the representation generated from the realization with the content of the original representation in beforehand.

The hyper links are stored in a hyper document and are mainly used for performing search operations in audio data equivalent to those which are possible in representation data which thus enables an improved access to the realization, e.g. via audio databases.

The mechanism disclosed in the precited U.S. Patent Application will now be discussed in more detail referring to Fig. 1. The mechanism uses an aligner 100 which comprises a structural analyzer 103 with input means. The structural analyzer 103 is connected via two output means to a time aligner 108 and a link generator 110. The aligner 100 further comprises a temporal analyzer 106 with input means. The temporal analyzer 106 is connected via output means to the time aligner 108. The time aligner 108 with two input means for receiving data from the structural analyzer 103 as well as from the temporal analyzer 106 is connected via output means to the link generator 110. The link generator 110 with two input means for receiving data from the structural analyzer 103 as well as from the time aligner 108 has an output means for sending data.

As illustrated in Fig. 1, the structuring process starts with a representation 101 and a realization 102. Usually both the representation 101 and the realization 102 are stored in a separate file, but each of the data sets may actually be distributed among several files or be merged in one complex hyper-media file. Alternatively, both representation 101 and the realization 102 may be fed into the system as a data stream.

The representation 101 is a descriptive mark-up document, e.g. the textual representation of a book, or the score of a symphony. An example of a realization 102 is an audio stream in an arbitrary format, e.g. WAVE or MPEG.

An examplary procedure for combining a representation 101 and a realization 102 of a multimedia stream is illustrated in Fig. 2. In a first processing step 201, the representation 101 is fed into the structural analyzer 103. The structural analyzer 103 analyzes the representation 101 and separates the original plain representation 104 and a structural information 105. The plain representation 104 includes the plain content of the representation 101, that is the representation 101 stripped of all the mark-up.

In step 202 of Fig. 2, which may be carried out before, after or at the same time as step 201, the realization 102, e.g. the audio stream, is fed into the temporal analyzer 106. The temporal analyzer 106 generates a time-stamped (or otherwise marked) representation 107 from the realization 102. It is advantageous to generate a time-stamped representation 107 of the complete realization 102. However, some embodiments create

marked or time-stamped representations 107 only of parts of the realization 102.
The time-stamped representation 107 includes the transcript and time-stamps of all elementary representational units like e.g. word or word clusters. In the above example a speech recognition engine is used as temporal analyzer 106 to generate a raw time-tagged transcript 107 of the audio file 102. Many commercially available speech recognition engines might be used, for example IBM's ViaVoice. However, in addition to the recognition of words, the temporal/marker analyzer 106 should be able to allocate time stamps and/or marks for each word.

In Fig. 2, step 203, the plain representation 104 derived from step 201 and the time-stamped representation 107 derived from step 202 are fed to the time aligner 108. The time aligner 108 aligns the plain representation 104 and the time-stamped representation 107. Thereby for the aligned elements, the time locator from the time-stamped representation 107 is attached to the content elements (e.g. words) from the plain representation 104 leading to the time-stamped aligned representation 109. The time aligner 108 creates an optimal alignment of the words from the time-stamped representation 107 and the words contained in the plain representation 104. This can be done by a variety of dynamic programming techniques.

In step 204 of Fig. 2, the structural information 105 and the time-stamped aligned representation 109, e.g. in form of data streams, are fed into a link generator 110. The link generator 110 then combines the locators of each element from the structural information 105 with a respective time locator from the time-stamped aligned representation 109, thereby creating connections between equivalent elements of representation 101 and realization 102, so called time-alignment hyper links 111. In an embodiment these hyper links 111 are stored in a hyperlink document. In an alternative embodiment these hyperlinks are transferred to a data base.

Now referring to Fig. 3, the basic concept of the method according to the present invention is illustrated by way of a high-level flow diagram. Starting with a realization 300, in the present example a video stream including an audio track or audio substream (e.g. a common movie) in English language, and a corresponding first representation 'A' 310 which is a plain text of the original audio substream, are synchronized 320 by the above described mechanism according to the prior art. The first represention A can already be used as a subtitle during presentation of the video/audio stream. It is further presumed that another representation B 340 for the audio substream exists which, in the present example, is a French translation of the original English audio stream or the corresponding English plain text (=representation 'A'), respectively. Synchronization 320 of the realization 300 and the first representation A 310 reveals a synchronized English version 355 of the original video/audio stream 300.

An example of such a synchronisation expressed by HyTime links(ISO 10744) is shown in Fig. 22. Leaving the technicalities that are specified in the document type definition linkweb.dtd aside, the output estabishes links of type primaryLink between a representation and a realisation by specifying the ids of the respective linkends. The location of the linkend for the representation (in the example a text file) is identfied by the id as used in the link and specified by a treelocator. The location of the linkend for the realisation (in the example an audio file in WAVE format) is identified by the ID as used in the link and specified by an urllocator giving the name of the file and start and endtime of the specified location.

It is emphasized that the video/audio stream is only an exemplary embodiment and can also be a pure audio stream for which translations can be provided too. In order to separate the above mechanism according to the prior art from the mechanism proposed by the invention, a dotted line 330 is used.

It is now assumed that the second representation B 340, which e.g. is a plain text representing the translation of the English plain text 310 into French, shall be synchronized too with the realization 300 in order that it can be used as a further time-synchronous subtitle when presenting the video/audio stream 300.

For both the representation A 310 and the representation B 340, at first the structure is analyzed and the two revealed structures are synchronized 360 with each other in order to build structure links between equivalent structural elements of both representations A and B 310, 340. The output 365 of step 360 is illustrated in Fig. 23. The so-called structurers are modules that extract a document structure appropriate for the media type from the data. The mechanisms to extract this structural information are different for different input data like e.g. video, audio, or text and are not part of this invention. Based on the links in 355 and 365, the synchronization 320 can be used to provide a time-synchronous alignment also between the representation B 340 and the video/audio stream 300 by merging both one-to-one links into common one-to-many links 375 and thus by inheriting the time information from representation A onto equivalent elements of representation B. These links are illustrated in Fig. 24.

In Figs. 4 and 5, for two different embodiments, the core system elements needed for the automatic generation of multi-lingual subtitles and their dependencies are depicted.

Referring to Fig. 4, as described above, the essential input data is a digital representation 400 of the audio signal associated with the material like e.g. a recording. In cases were the material is only available in analog formats, a digital representation has to be generated by known techniques. If the material to be sub-titled is a movie, additional visual data are available.

The box "Video" 410 in the overview refers to a digital encoding of the raw video data, i.e. the visual information without the audio tracks. Like for the audio signal 400, a digital encoding of analog video information and a separation of video and audio tracks is achieved by processing means according to the prior art. In many cases the audio-visual material is accompanied by a transcript 420 and/or a translation 430 thereof giving one or more textual representations 420, 430 of the spoken parts of the audio track 400. Like in the case of the audio-visual data it is assumed that this information is available in a digital format. Whenever only a typo-script is available, the textual information can be digitized by state of the art OCR technologies.

The realizations 400, 410 and the representations 420, 430 in the following are analyzed by means of corresponding structurers 440 - 470 in order to obtain structure information. Some structure can be inferred from most media data: A video 410 is often assembled by joining sequences of uninterrupted camera output. An interview can be structured by segmenting it into the intervals corresponding to a single speaker, and the the utterances of a speaker can be segmented into phrases. A set of structurers is used to extract this information. If no transcript 420, 430 is available, the structure of the audio 400 can be used to guide the automatic transcript generation by speech recognition. In a similar way the structure of the transcript 420, 430 can be used to guide the automatic translation in those cases where only one transcript 420 is available.

In the next step two classes of aligners 480, 485 are used to synchronize both structures and data. Structure aligners 480 build a unified structure from the different input structures and a content aligner 485 synchronizes the master realization (usually the audio track) with the master representation (usually the transcript in the original language). The result of this alignment step is a web of relations between different versions of the content and different structural views of the content.

In the next step a link generator 490 extracts a single view from this web of relationships by discarding some of the synchronization information. For instance, in the case of the alignment of a transcript with a multitude of translations, only the information needed for an individual target language is selected from the alignment. This view is used by a renderer 500 to actually drive the generation of the synchronized version by reformatting these data in a form appropriate for a sub-titling engine or a speech synthesis system.

Referring now to Fig. 5, a second embodiment is shown that describes the application of the invention in the case where only an audio recording or a video with sound track is available. Since in this case no representational information is available, both the text spoken on the audio track as well as the translation of the text have to be generated automatically.

To derive the representational information, the audio track is passed through a structurer that separates the audio into sentences. For the further details of that procedure it is referred to European Patent Application ____________ (docket no. DE9-2000-0060 of the present applicant) which is regarded as fully incorporated herein by reference. According to that procedure, a digitized speech signal is input to an F0 (fundamental frequency) processor that computes a continuous F0 data from the speech signal. By the criterion voicing state transition (voiced/unvoiced transitions the speech signal is presegmented into segments. For each segment it is evaluated whether F0 is defined or not defined (i.e. F=ON/OFF). INn case of a not defined F0 (i.e. F0 = OFF) a candidate segment boundary is assumed as described above and, starting from that boundary, prosodic features are computed. The feature values are input into a classification tree and each candidate segment is classified thereby revealing, as a result, the existence or non-existence of a semantic or syntactic speech unit.

Like in the first embodiment the output of this structurer is passed on to the aligner but in addition the structure derived is used to guide a state-of-the-art speech recognition system. Current speech recognizers do not automatically transcribe the punctuation that is used to separate sentences in the transcript but require a speaker to explicitly insert such tags. This information is taken from the audio structure and the output of the transcriber transformed from a stream of words into a sequence of sentences. This plain text is forwarded to a text structurer as described in the first embodiment and together with text structure fed into a state-of the art machine language translation system. Like in the first embodiment, the translation produced by this system is fed into a structurer and into an aligner. Further processing of the data is similar to the first embodiment.

The first embodiment (Fig. 4) of the system according to the invention is now described in more detail referring to the two block diagrams depicted in Fig. 6a und 6b. Referring at first to an exemplary audio structurer depicted in Fig. 6a, the audio data 600 are fed into an audio structurer 610 that separates speech 620 and non-speech segments 630 and passes the speech segments 620 on to an aligner I 640 as described above and disclosed in U.S. Patent Application No. ____________ (docketno. DE9-1999-0053). It is noteworthy that only the speech segments 620 are input to the aligner I, whereas the non-speech segments 630 are not aligned since there is nothing to align. The mixed blocks 650 consisting of both speech and non-speech segments can be used on higher processing levels, e.g. generation of the tree locator 1 depicted in Fig. 10. The output of the aligner I 640 is a link file I 645 (not detailed in Fig.s 6a and b).

Now referring to Fig. 6b, the transcript 660 and translation 670 of the transcript are passed through appropriate text structurers 680, 690 which generate a document structure like that shown for the sample text "Donna Leon, Acqua alta", Macmillan 1996, depicted in Fig. 12. As described above, both structures are then processed by a structure aligner 700 (aligner II) that establishes the correspondence between equivalent structural elements and produces a link file II 710. The link file I 645 and link file II 710 are combined to a link file by a link generator 720 that is renderer by means of a renderer 730 as described above in Fig. 4.

The structuring techniques detailed in the following sections are provided by way of example to ease the understanding of the different aligners.

Structurers

1. Audio Structurer

An audio structure suitable for the automatic generation of subtitles derives a structure from the audio stream that segments the continuous audio stream containing speech into phrases and a super-structure as illustrated in Fig. 7: The root node 700 of the tree corresponds to the complete audio stream. The children of the root node represent continuous segments of speech (710) respectively non-speech (720) audio. Each segment containing recorded speech is segmented into continuous intervals containing the utterances of an individual speaker (730, layer 3) and the utterances of an individual speaker are segmented into phrases (roughly corresponding to sentences in written language) which are the leaf nodes 740 of the audio structure tree. State of the art techniques are available for the automatic segmentation of speech/non-speech parts and for the generation of segments corresponding to individual speakers (see for instance Lynn Wilcox, Francine Chen, Don Kimber, Vijay Balasubramanian, Segmentation of Speech Using Speaker Identification, Proc. ICASSP '94; Adelaide, Australia, 161-4, 1994; Claude Montacié, Marie-José Caraty, A Silence/Noise/Music/Speech Splitting Algorithm, Proc. ICSLP'98).

The output from the audio structurer is either stored or forwarded to aligners as a sequence of tree locators, as illustrated by the exemplary output depicted in Fig. 8. For details it is referred to the description of the preferred embodiments hereinbelow.

2. Text Structurer

The detail of structural information that can be extracted from a text depends on the markup convention used to write the text: A text may either be a plain text containing no more markup information than required by the orthographic rules of a language, a structured plain text that uses certain convention to mark the structure of the text document, or a text encoded in a formal markup language like e.g. HTML, XML or LaTex.

Plain text can be segmented by exploiting the typographic conventions of a language's writing system. Most languages use special typographic signs to mark sentence boundaries or the flag direct speech. During the last decades such conventions are used even in writing system that traditionally did not rely on such markup like e.g. Chinese or classical Latin. A structurer exploiting only such features is specific for the language of the text and exploits a fixed set of rules.

Structured plain text uses an additional set of conventions - that varies between different text sources and corpora - to denote the super-structure of the text like paragraphs or chapters. Example of such documents are the texts of the Gutenberg etext collection where an empty line is used to mark paragraph boundaries or documents conforming to the structuring requirements (see for instance DIN 1421) where a decimal scheme of tags is used to label titles according to structural position of their paragraphs. Structurers exploiting structured plain text depend on the structuring conventions applied and have to be written or adapted for each text corpus.

Thereupon, texts tagged using a standardized markup language can be viewed as instantiations of a document type either specified explicitly, like e.g. with document type definitions (DTDs) used by markup languages derived from SGML, or implicitly (like e.g with the style sheets or templates used by LaTex respectively Word). For such documents a document object model is derived from the formal definition of the document structure in e.g. the DTD and segments of the document are linked with a tree representing the document structure. Structurers for markuped texts depend on the markup language used but are otherwise generic.

For all three text types, the output of the text structurer is a sequences of tree locators that is stored or forwarded to the Aligners. Only the depth of the tree is different for the three text classes, usually it is flat for plain texts and more elaborate and multi-layered for texts tagged with a markup language.

Aligners

In addition, aligner modules are used to synchronize the different structures determined by the structurers with each other and with the realisation of the audio-visual stream. Both, structure-structure aligners and structure-content aligners compute the optimal match between their input data (usually by applying a dynamic programming algorithm) and produce as an output a sequence of links between the input data in the form of independent hyperlinks. There are various notations to express independent hyperlinks, the notational convention used here is the one of HyTime ilinks (see e.g. LIT ISO 10744).

A link generator assembles the dyadic hyperlinks generated by the aligners into a web of multi-ended links. The output of the link generator is a sequence of one-to-many links that connect all equivalent elements in the different realizations and representations. There are various notation to express independent hyperlinks, the notational convention used here is the one of HyTime ilinks (LIT ISO 10744).

Thereupon, the link-web collected by the link generator contains all the relationships between the different realizations and representations of a media stream. In most cases an application is not interested in all these data but only in a specific view of these relationship. A typical example for such a view is the link between the audiovisual data and the translation of the original dialogue in one target language. For that purpose, renderers are used. Hereby, in a first processing step, the renderer selects this view from the link-web by selecting only those link-ends from the one-to-many links that refer to the targets the application wants to make use of. In a second processing step, the selected view is rendered, i.e. transcoded into the format needed to make the selected link-ends visible or audible. In the simplest case, rendering consist only in reformatting the links and the referred link-ends into a format conforming to a presentation language like e.g. SMIL. In more complex cases, the renderer feeds a subtitling hardware with the information needed to produce the overlayed subtitles for the video, or a text-to-speech system to produce an audible synchronization.

In the following, a preferred embodiment of the proposed mechanism for the automatic generation of sub-titles will be described in more detail, i.e. a system where a transcript and a translation are available.

Referring back to the first embodiment depicted in Fig. 4, the audio data is fed into a structurer that uses state of the art techniques to segment the audio stream into speech and non-speech blocks (see for instance Lynn Wilcox, Francine Chen, Don Kimber, Vijay Balasubramanian, Segmentation of Speech Using Speaker Identification, Proc. ICASSP '94, Adelaide, Australia, 161-4, 1994 and Claude Montacié, Marie-José Caraty, A Silence/Noise/Music/Speech Splitting Algorithm, Proc.). The document structure derived by this structurer is shown in Fig. 9. Such a document structure is typical for most audio books that combine segments of music with narration: the document starts with a short piece of music as introduction, the individual chapters are separated by short musical interludes, and after the last narrative segment a final piece of music is used as a trailer. In the tree diagram representing the structure of the audio the node in layer one represents the complete audio data. Each node in layer two represents a homogeneous segment of either speech or non-speech data, the nodes in layer one are ordered from left to right according to increasing time. All nodes carry as attributes a label for the type of segment (M: mixed, N: non-speech, S:speech) and a reference to the boundaries of the segment in appropriate units like time-offset, or byte-offset into the audio data. Various methods exist to encode such an audio structure tree and communicate it to the aligner, one method is shown in Fig. 8: The first column of the table contains the tree locator to uniquely identify the node, the second column contains a label for the segment type, and column three and four are used to store start and stop time of the segment.

Both the transcript and the translation thereof are processed by a text structure analyzer the output of which is a stream of tree locators expressing the structure of the text and a text decorated with markup. The processing steps for a plain text like the one shown in Fig. 12 are illustrated by way of the flowchart depicted in Fig. 10. The file or stream containing the text is opened 1000 and in a first step the language of the text is determined 1010. This can be done either interactively by an operator or by automatic methods as described in Y.K. Muthusamy, E. Barnard and R. A. Cole, "Automatic Language Identification: A Review/Tutorial," IEEE Signal Processing Magazine, October 1994 I.D. Melamed, A Geometric Approach to Mapping Bitext Correspondence, Proceedings EMNLP'96, Philadelphia 1996. Then the language-specific table of tags to encode the sentence structure of the text and the corpus-specific table of tags for paragraph structure are loaded 1020. In the example texts shown in Fig. 12 both languages use the same end-of-sentence tag, a full-stop, but different tags to mark direct speech (double quotes in German versus single quotes in English). In both texts an empty line is used to mark paragraph boundaries. As first processing step the parser initiates the processing and stores or forwards start tag for all structural elements used. Until the complete text file or text stream is processed, the parser tests 1040 each character first whether it is an end-of-sentence marker and then whether it is an end-of-paragraph marker 1050. Whenever an end-of-sentence character is found, an end-of-sentence tag is added to the text, a new tree-locator computed and tree-locator and offset are either stored or forwarded to the aligner 1060. Whenever an end-of-paragraph character is found 1070, an end-of-paragraph tag is added to the text, a new tree-locator computed and tree-locator and offset are either stored or forwarded to the aligner 1080. After the complete text is processed 1090, closing tags for all open tags are written to the text 1100.

The resulting structured text is shown in Fig. 13. The document structure of the example text depicted in Fig. 12 is shown in Fig. 16 where, instead of word nodes, the number of word children is shown in the sentence nodes. For a text structured in a mark-up language, additional superstructure grouping the sentences into paragraphs, chapters etc. is extracted from the mark-up with methods as described in precited U.S. Patent Application No. ________________(docket DE9-1999-0053). Various methods exist to encode such a text structure tree and to relate it to the text, the one used in the first embodiment is similar to the encoding described in precited U.S. Patent Application No. ________________ (docket DE9-1999-0053) using mapping tables between tree-locators and node-names (Fig. 14) and mapping tables between tree locators and the text (Fig. 15). Audio, audio structure, text structures and texts are passed into the aligner and processed by two alignment modules.

The speech segments of the audio, as identified by the audio structure shown in Fig. 9, are aligned with the transcript of the original language using the methods described in the above cited U.S. Patent Application No. ______________. The second alignment module aligns the structure of the original text and the translation and produces a link file that links corresponding nodes in both structures, as shown is Fig. 18. As can be seen from the example text, translations usually preserve the structure of the original text, even in cases where they are slightly uncomplete: The phrase "pronouncing her last name in the Italian fashion" is missing from the German version of the text but the structure of both examples is identical. Therefore the alignment of the documents structure implies in most cases an alignment of text and translation. Of course, structural equivalence can not guarantee perfect alignment: Cases where, e.g. one sentence is split into two sentences in the translation and two other sentences are merged will not align correctly based on structural information only. In cases where such transformations between text and translation are likely (like in the case of German translated into English), or in cases where the structure aligner produces too many mismatches, the structure based text alignment module can be amended by a bitext aligner.

The structure aligner is illustrated by way of a flow chart shown in Fig. 11: The document structures for both, the original text and the translation are fed 1200 into module 1210 that computes the maximum agreement sub-tree (MAST), i.e. an optimal alignment of both tree structures. This is done by known techniques like e.g. the algorithms described in Martin Farach / Mikkel Thorup, Sparse Dynamic Programming of Evolutionary-Tree Comparison, SIAM J. Comput, 26, 210-30 (1997) or the algorithms as described in the literature cited therein. If the document structure of the translation is equal 1220 with the document structure of the text, i.e. if all nodes of the translation tree align with the corresponding nodes of the text tree, the links between both document structures are stored or forwarded 1230 to the link generator (see also Fig. 19). If both document structures are not equal 1220 , further processing depends on the percentage of unaligned tree nodes: If the percentage of unaligned nodes is below a user-selectable threshold 1240 and the system is not in interactive mode 1250, the links between both document structures are stored or forwarded 1260 to the aligner, leaving some elements of the document structure unaligned, i.e. There will be elements in the text or the translation that do not have counterparts in other document. If the system is in interactive mode, the unaligned structure element and the text instantiating them is displayed 1270 in a GUI together with the preceding and succeeding structure elements and their texts. This allows an operator to manually correct the alignment. The corrected alignment is stored or forwarded 1280 to the link generator. If the percentage of unaligned nodes in the document structure is above the user-selected threshold, the markuped text and the markuped translation are aligned 1290 using state-of-the-art techniques like the one described in I.D. Melamed, A Geometric Approach to Mapping Bitext Correspondence, Proc. of the First Conference on Empirical Methods in Natural Language Processing, Philadelphia

1996 or M. Simard / P. Plamondon, Bilingual sentence alignment, Proc. of the Second Conference of the Association for Machine Translation in the Americas, Montreal 1996. Again, the links between both document structures resulting 1300 from the bitext alignment are stored or forwarded 1310 to the link generator.

As can be seen in Fig. 20, the link information from both alignment modules is combined into one web of one to many links by the link generator. As shown in the flow diagram depicted in Fig. 19, the link generator collects for markup language ID 1500 in the text document structure (Fig. 14) all the links 1530, 1540 that reference this node and writes 1550 the link-web that is stored or forwarded to the renderer.

The last two processing steps are performed by the renderer: In a generic step, the renderer generates a view from the link-web by selecting the active link-ends from the link-web, e.g. the language selected and the audiovisual data, and follows the links to their targets, as can bee seen in Fig. 21. In a second step, this information is either formatted according to the grammar of the presentation language or synthesized to become visible or audible.

CLAIMS

1. A computer-based method of synchronizing a realization of a media stream, wherein having a first representation already synchronized with the realization, with at least one second representation, comprising the steps of:

determining structure information for the first representation and the at least one second representation;

determining structure association information between the first representation and the at least one second representation;

inheriting the already existing synchronization between the realization and the first representation to the at least one second representation by means of the structure association information.

2. Method according to claim 1, wherein analyzing the structure, in particular the text structure, of the first and the at least one second representation and providing a stream of tree locators.

3. Method according to claim 2, wherein aligning the at least two determined structures.

4. Method according to claim 2 or 3, wherein aligning the content of the realization with the first representation.

5. Method according to claim 3 or 4, wherein providing a web of relations between different versions of the content and/or different structural views of the content.

6. Method according to any of claims 3 to 5, wherein aligning an audio stream with a corresponding audio structure and/or a text stream with a corresponding text structure.

7. A system for synchronizing a realization of a media stream, wherein having a first representation already synchronized with the realization, with at least one second representation, comprising

a first structurer for providing structure information for the first representation;

at least a second structurer for providing structure information for the at least second representation;

a first aligner for aligning the at least two provided structure informations.

8. System according to claim 7, further comprising at least one renderer for rendering the at least one synchronized second representation in a form suitable for displaying, in particular as an overlayed subtitle.

9. System according to claim 8, further comprising a tree aligner for determining a tree structure for the media stream.

10. System according to any of claims 7 to 9, further comprising means for detecting speech-/non-speech boundaries and/or transitions and/or speaker changes.

11. A data processing program for execution in a data processing system comprising software code portions for performing a method according to any of claims 1 to 6 when said program is run on said computer.

12. A computer program product stored on a computer usable medium, comprising computer readable program means for causing a computer to perform a method according to any of claims 1 to 6 when said program is run on said computer.

ABSTRACT

Disclosed is a method and system for computerized synchronizing an audio stream, for which a first synchronized textual representation usable for subtitling of the audio stream in the original language is existing, with a second synchronized textual representation which can be used as an alternative subtitle e.g. comprising a transcription of the original language into another language.

The proposed method and system accomplish the foregoing by building time-synchronous links between the audio stream and, for instance, the textual representations of the words spoken in the audio stream. More particularly, synchronization between the audio stream and the first representation is used to inherit the synchronization information to the second representation wherein using structure association information determined between the first and the second representation.
(Fig. 3)

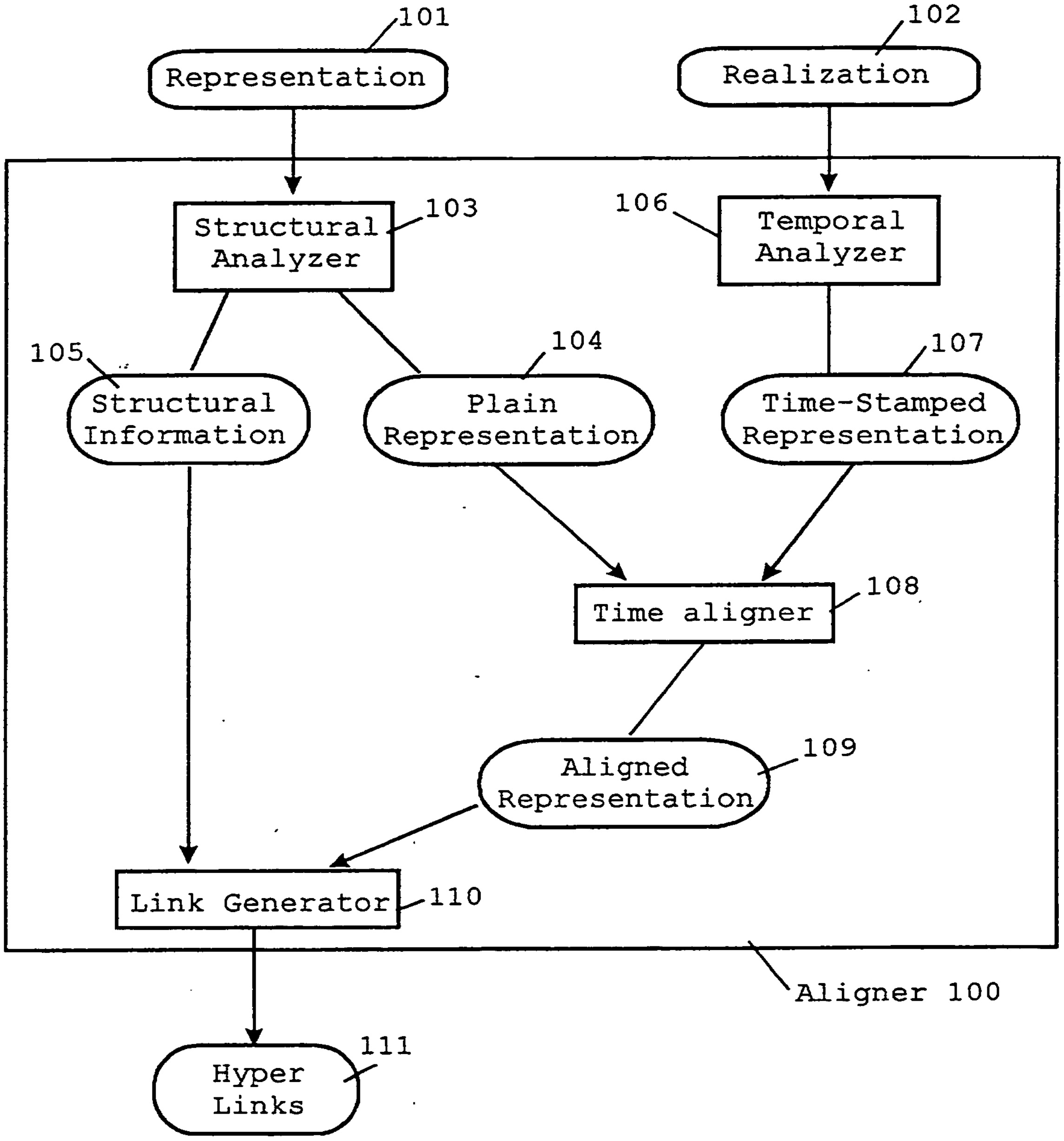
101
Representation
102
Realization
Structural Analyzer
103
106
Temporal Analyzer
105
Structural Information
104
Plain Representation
107
Time-Stamped Representation
Time aligner
108
Aligned Representation
109
Link Generator
110
Aligner 100
Hyper Links
111

FIG. 1

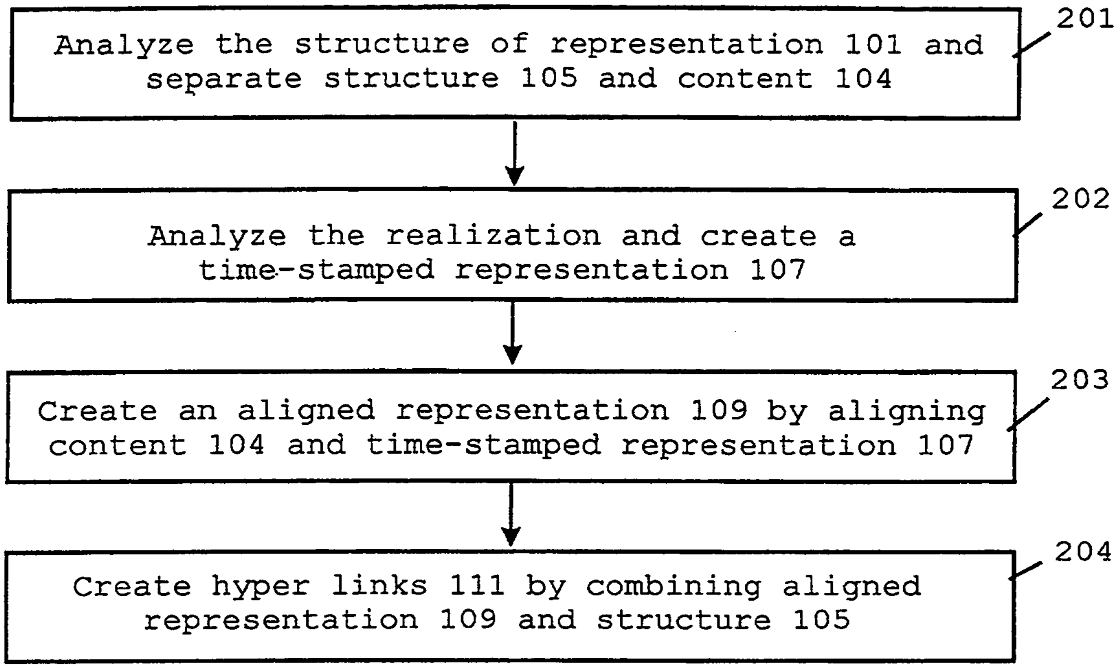
Analyze the structure of representation 101 and separate structure 105 and content 104
201
Analyze the realization and create a time-stamped representation 107
202
Create an aligned representation 109 by aligning content 104 and time-stamped representation 107
203
Create hyper links 111 by combining aligned representation 109 and structure 105
204

FIG. 2

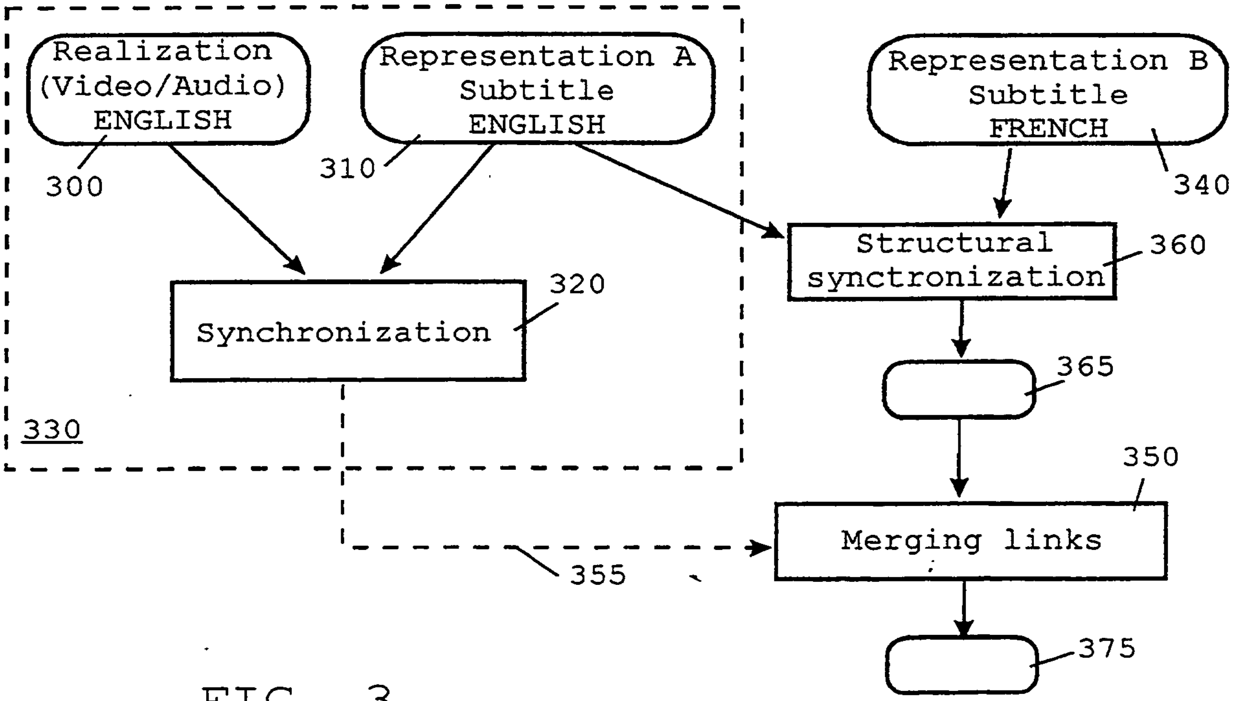
Realization (Video/Audio) ENGLISH
300
Representation A Subtitle ENGLISH
310
Representation B Subtitle FRENCH
340
Structural synctronization
360
Synchronization
320
330
365
350
Merging links
355
375

FIG. 3

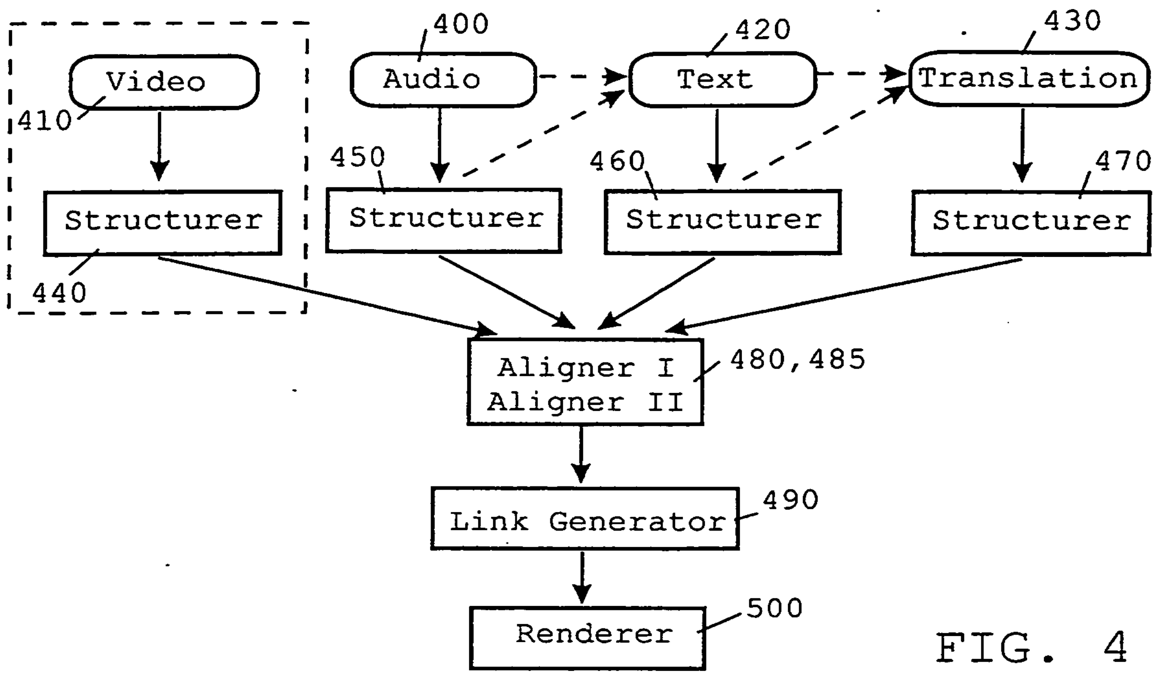
Video
410
Structurer
440
400
Audio
450
Structurer
420
Text
460
Structurer
430
Translation
470
Structurer
Aligner I
Aligner II
480,485
Link Generator
490
Renderer
500

FIG. 4

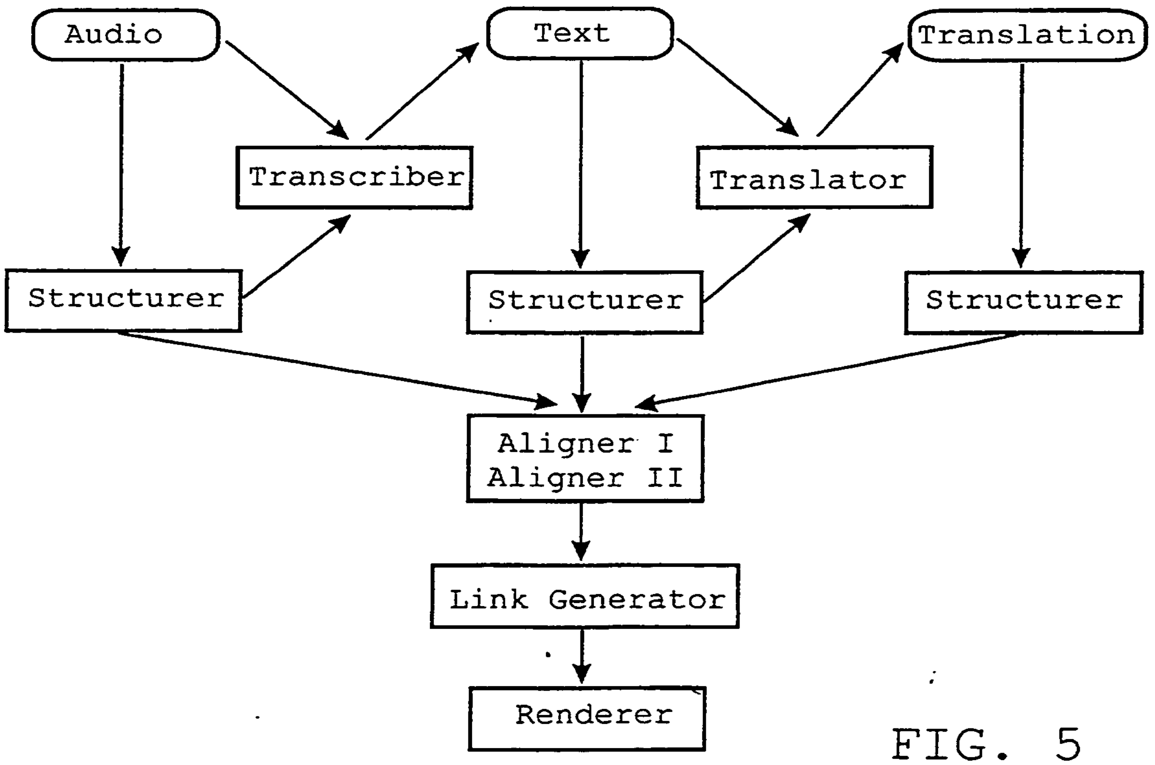
Audio
Transcriber
Text
Translator
Translation
Structurer
Structurer
Structurer
Aligner I
Aligner II
Link Generator
Renderer

FIG. 5

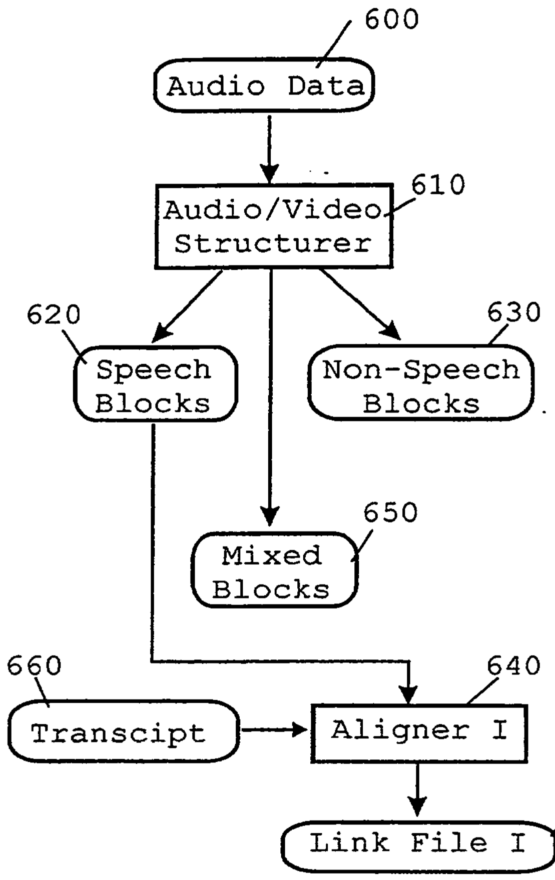
600
Audio Data
610
Audio/Video Structurer
620
Speech Blocks
630
Non-Speech Blocks
650
Mixed Blocks
660
Transcipt
640
Aligner I
645
Link File I

FIG. 6A

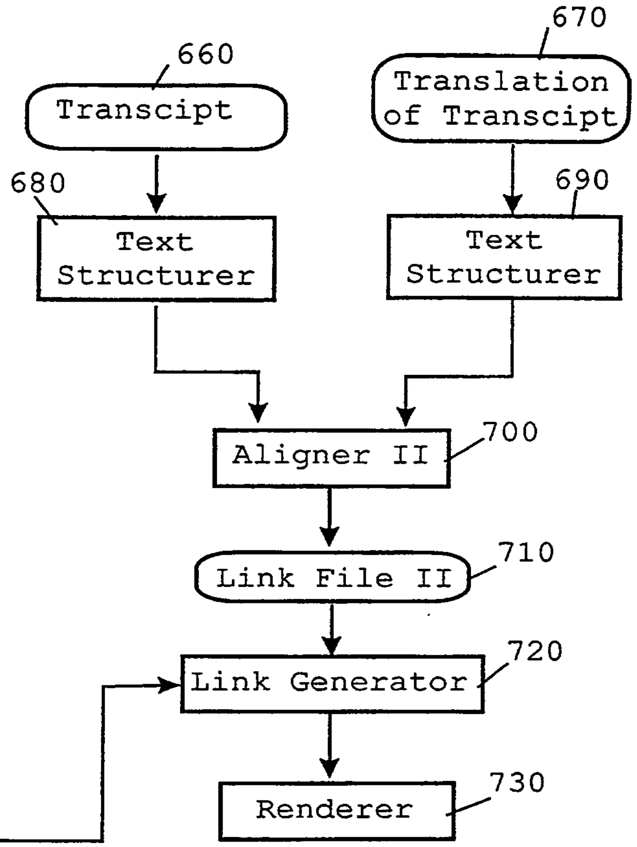
660
Transcipt
670
Translation of Transcipt
680
Text Structurer
690
Text Structurer
700
Aligner II
710
Link File II
720
Link Generator
730
Renderer

FIG. 6B

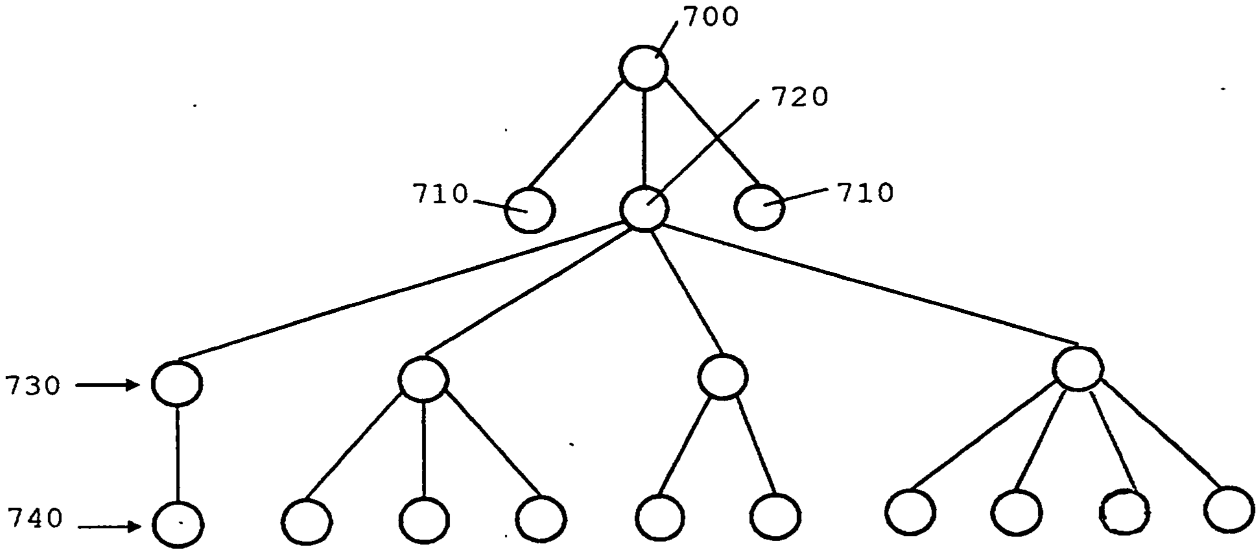

FIG. 7

Tree-locator	Segment Type	Offset Start	Stop
1	M	0:00	10:00
1 1	N	0:00	0:20
1 2	S	0:20	9:40
1 3	N	9:40	10:00
1 2 1	1	0:20	0:45
1 2 2	2	0:45	2:30
1 2 3	1	2:30	3:25
1 2 4	2	3:25	10:00
1 2 1 1	P	0:20	0:45
1 2 2 1	P	0:45	1:10
1 2 2 2	P	1:10	1:55
.....			

FIG. 8

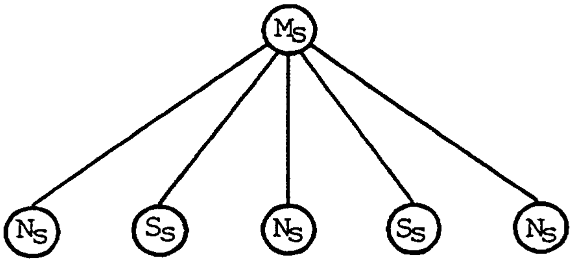

FIG. 9

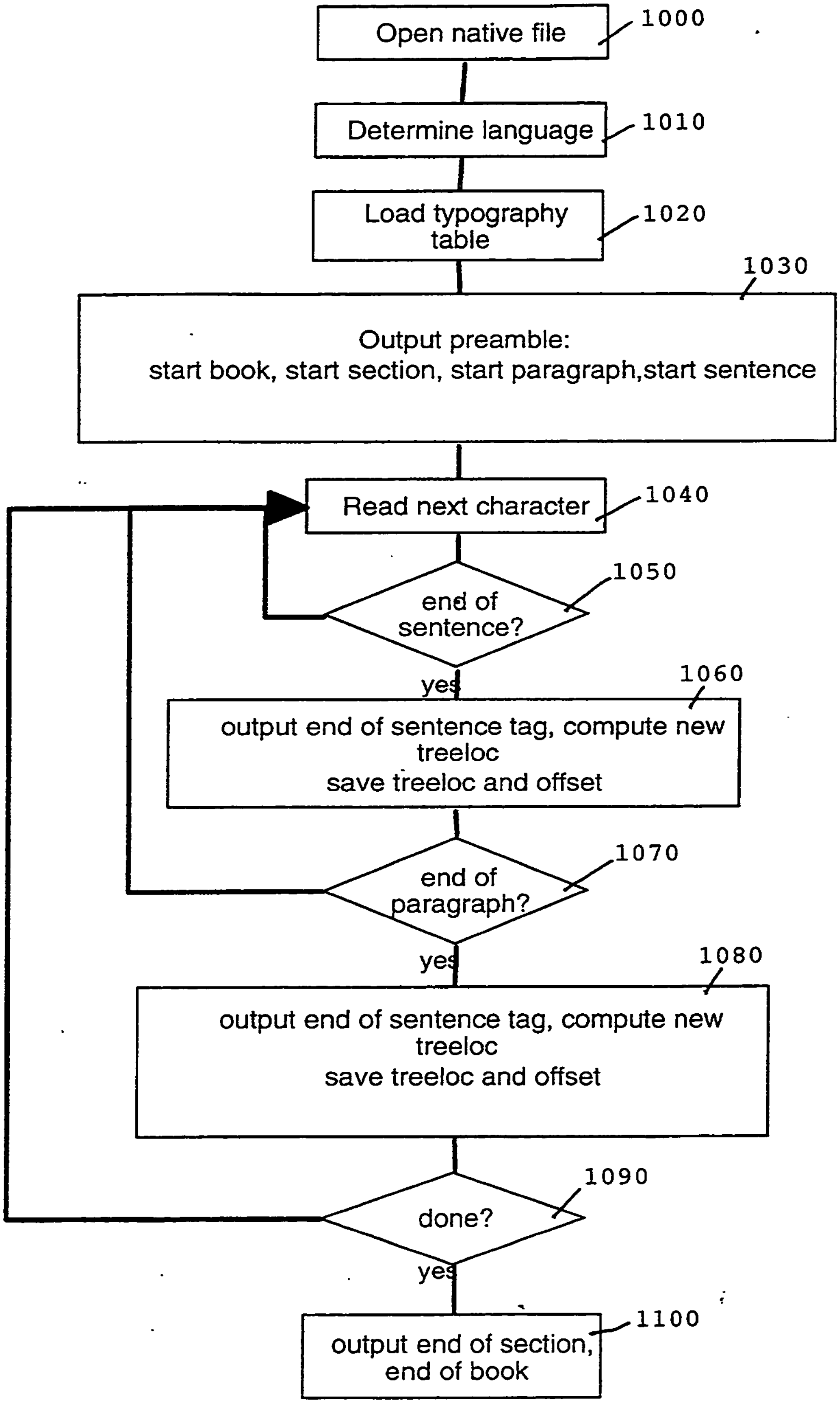
Open native file
1000
Determine language
1010
Load typography table
1020
1030
Output preamble:
start book, start section, start paragraph,start sentence
Read next character
1040
end of sentence?
1050
yes
1060
output end of sentence tag, compute new treeloc
save treeloc and offset
end of paragraph?
1070
yes
1080
output end of sentence tag, compute new treeloc
save treeloc and offset
done?
1090
yes
output end of section, end of book
1100

FIG. 10

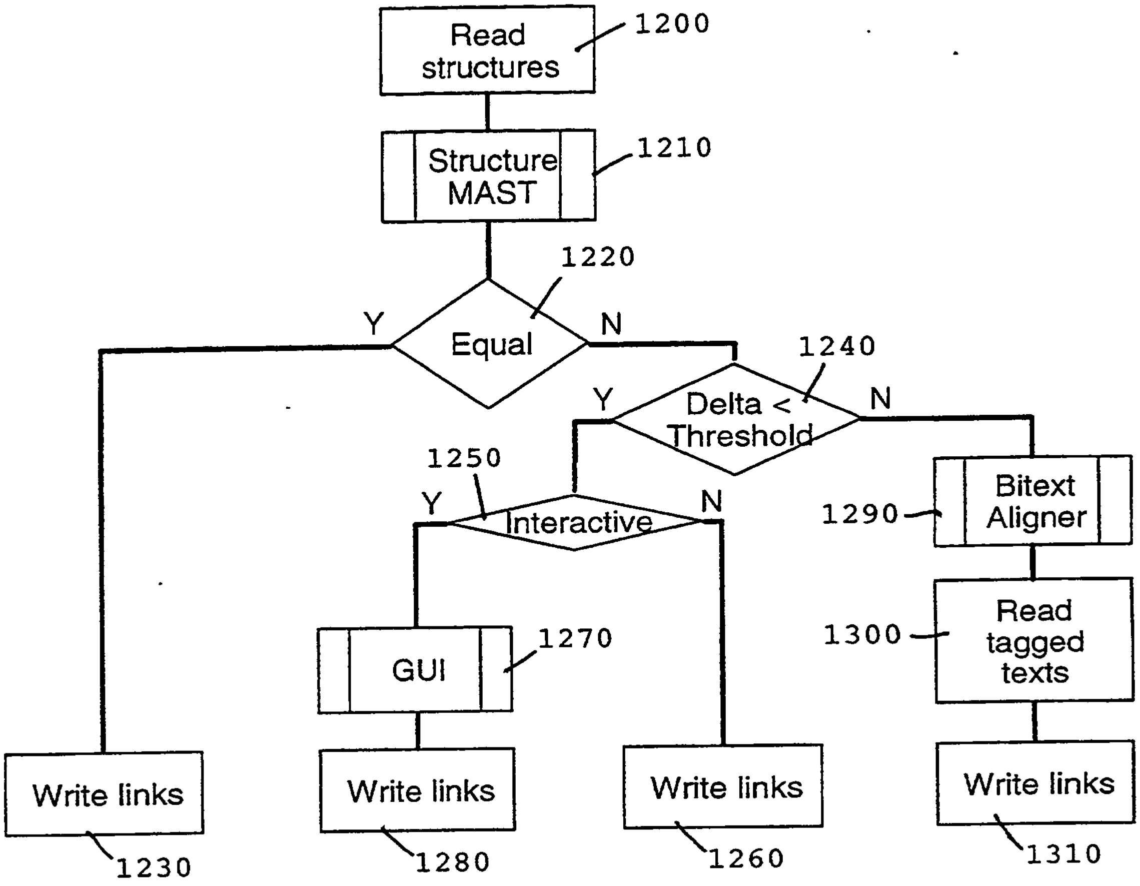

FIG. 11

The first one, short and fair-
haired, smiled back and started up the last flight of steps.
His companion, taller and darker, took a deep breath, then came up behind him.
When the first man got to the door, he paused and waited for the other to join him.

'Dottoressa Lynch?' the blond one asked, pronouncing her last name in the
Italian fashion.

'Yes,' she answered, stepping back from the door to allow them to enter.

Der erste, klein und blond, laechelte zurueck und betrat das letzte Treppenstueck.
Sein Begleiter, groeszer und dunkler, holte noch einmal tief Luft und kam ihm nach.
Als der erste Mann bei der Tuer angelangt war, blieb er stehen und wartete
auf den anderen.

"Dottoressa Lynch?" fragte der Blonde

"Ja", antwortete sie, indem sie zuruecktrat, um sie hereinzulassen.

FIG. 12

```
<!DOCTYPE book SYSTEM "book.dtd" [
]>
<book>
<chapter>
<section>
<p>
<s>The first one, short and fair-
haired, smiled back and started up the last flight of steps.</s>
<s>His companion, taller and darker, took a deep breath, then came up behind him.</s>
<s>When the first man got to the door, he paused and waited for the other to join him.</s>
</p>
<p>
<s>’Dottoressa Lynch?’ the blond one asked, pronouncing her last name in the Ital-
ian fashion.</s>
<s>’Yes,’ she answered, stepping back from the door to allow them to enter.</s>
</p>
</section>
</chapter>
</book>

<!DOCTYPE book SYSTEM "book.dtd" [
]>
<book>
<chapter>
<section>
<p>
<s>Der erste, klein und blond, laechelte zurueck und betrat das let-
zte Treppenstueck.</s>
<s>Sein Begleiter, groeszer und dunkler, holte noch ein-
mal tief Luft und kam ihm nach.</s>
<s>Als der erste Mann bei der Tuer angelangt war, blieb er ste-
hen und wartete auf den anderen.</s>
</p>
<p>
<s>"Dottoressa Lynch?" fragte der Blonde</s>
</p>
<p>
<s>"Ja", antwortete sie, indem sie zuruecktrat, um sie hereinzulassen.</s>
</p>
</section>
</chapter>
</book>
```

FIG. 13

Tree Locator	SGML Id
1 1 1 1 1 1	ldeutsch1
1 1 1 1 2 1	ldeutsch2
1 1 1 1 3 1	ldeutsch3
1 1 1 2 1 1	ldeutsch4
1 1 1 3 1 1	ldeutsch5

Tree Locator	SGML Id
1 1 1 1 1 1	leng1
1 1 1 1 2 1	leng2
1 1 1 1 3 1	leng3
1 1 1 2 1 1	leng4
1 1 1 3 1 1	leng5

FIG. 14

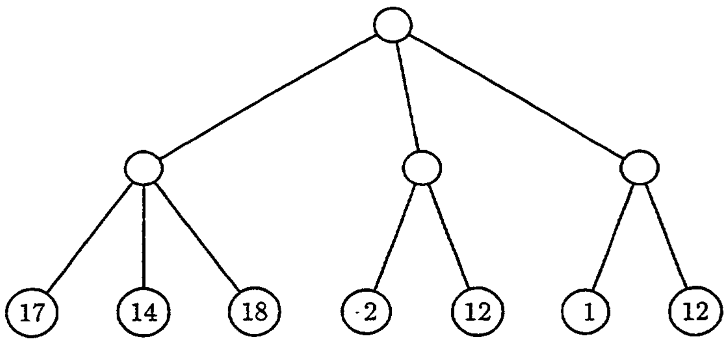

FIG. 16

Tree locator	Text
1 1 1 1 1 1	The first one, short and fair-haired, smiled back and started up the last flight of steps.
1 1 1 1 2 1	His companion, taller and darker, took a deep breath, then came up behind him.
1 1 1 1 3 1	When the first man got to the door, he paused and waited for the other to join him.
1 1 1 2 1 1	'Dottoressa Lynch?' the blond one asked, pronouncing her last name in the Italian fashion.
1 1 1 3 1 1	'Yes,' she answered, stepping back from the door to allow them to enter.

Tree locator	Text
1 1 1 1 1 1	Der erste, klein und blond, laechelte zurueck und betrat das letzte Treppenstueck.
1 1 1 1 2 1	Sein Begleiter, groeszer und dunkler, holte noch einmal tief Luft und kam ihm nach.
1 1 1 1 3 1	Als der erste Mann bei der Tuer angelangt war, blieb er stehen und wartete auf den anderen.
1 1 1 2 1 1	"Dottoressa Lynch?" fragte der Blonde
1 1 1 3 1 1	"Ja", antwortete sie, indem sie zuruecktrat, um sie hereinzulassen.

FIG. 15

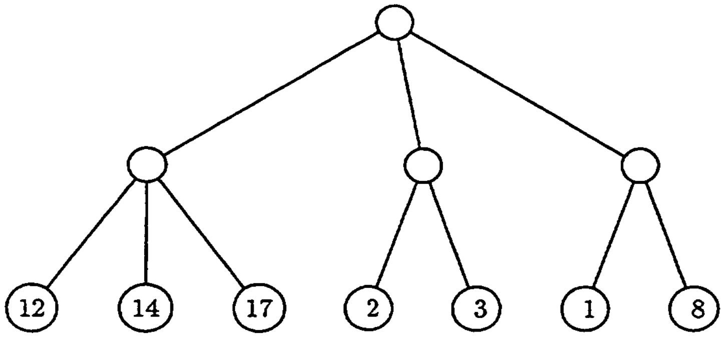

FIG. 17

```
<!DOCTYPE linkweb SYSTEM "linkweb.dtd" [
  <!ENTITY elink SYSTEM "leone.sgm" CDATA SGML>
  <!ENTITY dlink SYSTEM "leond.sgm" CDATA SGML>
]>

<linkweb>

<audio linkends="leng1 ldeutsch1">
<treeloc id="leng1"> locsrc=elink>1</treeloc>
<treeloc id="ldeutsch1" locsrc=dlink>1</treeloc>

<audio linkends="leng2 ldeutsch2">
<treeloc id="leng2"> locsrc=elink>1 1</treeloc>
<treeloc id="ldeutsch2" locsrc=dlink>1 1</treeloc>

<audio linkends="leng3 ldeutsch3">
<treeloc id="leng3"> locsrc=elink>1 1 1</treeloc>
<treeloc id="ldeutsch3" locsrc=dlink>1 1 1</treeloc>

<audio linkends="leng4 ldeutsch4">
<treeloc id="leng4"> locsrc=elink>1 1 2</treeloc>
<treeloc id="ldeutsch4" locsrc=dlink>1 1 2</treeloc>

<audio linkends="leng5 ldeutsch5">
<treeloc id="leng5"> locsrc=elink>1 1 3</treeloc>
<treeloc id="ldeutsch5" locsrc=dlink>1 1 3</treeloc>

<audio linkends="leng6 ldeutsch6">
<treeloc id="leng6"> locsrc=elink>1 2</treeloc>
<treeloc id="ldeutsch6" locsrc=dlink>1 2</treeloc>

<audio linkends="leng7 ldeutsch7">
<treeloc id="leng7"> locsrc=elink>1 2 1</treeloc>
<treeloc id="ldeutsch7" locsrc=dlink>1 2 1</treeloc>

<audio linkends="leng8 ldeutsch8">
<treeloc id="leng8"> locsrc=elink>1 2 2</treeloc>
<treeloc id="ldeutsch8" locsrc=dlink>1 2 2</treeloc>

....
```

FIG. 18

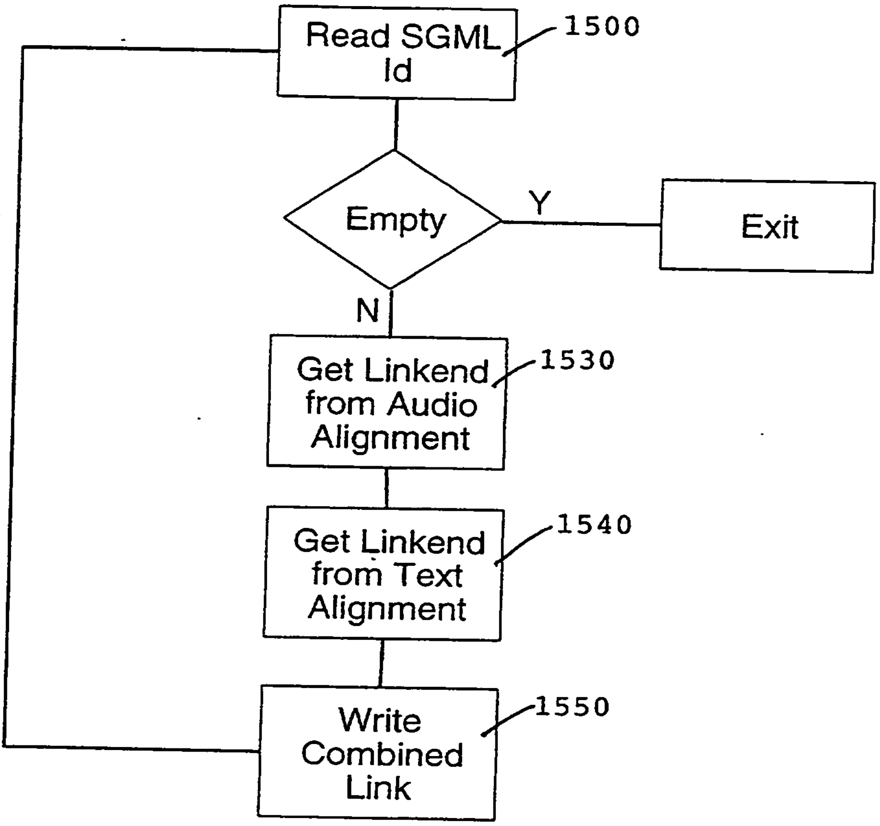
Read SGML Id
1500
Empty
Y
Exit
N
Get Linkend from Audio Alignment
1530
Get Linkend from Text Alignment
1540
Write Combined Link
1550

FIG. 19

```
<!DOCTYPE linkweb SYSTEM "linkweb.dtd" [
  <!ENTITY elink SYSTEM "leone.sgm" CDATA SGML>
  <!ENTITY dlink SYSTEM "leond.sgm" CDATA SGML>
]>

<linkweb>

<audio linkends="leng1 ldeutsch1 audio1">
<treeloc id="leng1"> locsrc=elink>1</treeloc>
<treeloc id="ldeutsch1" locsrc=dlink>1</treeloc>
<urlloc id="audio1">file=aquca.wav start=xxx end=yyy unit=ms</urlloc>

<audio linkends="leng2 ldeutsch2 audio2">
<treeloc id="leng2"> locsrc=elink>1 1</treeloc>
<treeloc id="ldeutsch2" locsrc=dlink>1 1</treeloc>
<urlloc id="audio2">file=aquca.wav start=xxx end=yyy unit=ms</urlloc>

<audio linkends="leng3 ldeutsch3 audio3">
<treeloc id="leng3"> locsrc=elink>1 1 1</treeloc>
<treeloc id="ldeutsch3" locsrc=dlink>1 1 1</treeloc>
<urlloc id="audio3">file=aquca.wav start=xxx end=yyy unit=ms</urlloc>

<audio linkends="leng4 ldeutsch4 audio4">
<treeloc id="leng4"> locsrc=elink>1 1 2</treeloc>
<treeloc id="ldeutsch4" locsrc=dlink>1 1 2</treeloc>
<urlloc id="audio4">file=aquca.wav start=xxx end=yyy unit=ms</urlloc>

<audio linkends="leng5 ldeutsch5 audio5">
<treeloc id="leng5"> locsrc=elink>1 1 3</treeloc>
<treeloc id="ldeutsch5" locsrc=dlink>1 1 3</treeloc>
<urlloc id="audio5">file=aquca.wav start=xxx end=yyy unit=ms</urlloc>

<audio linkends="leng6 ldeutsch6 audio6">
<treeloc id="leng6"> locsrc=elink>1 2</treeloc>
<treeloc id="ldeutsch6" locsrc=dlink>1 2</treeloc>
<urlloc id="audio6">file=aquca.wav start=xxx end=yyy unit=ms</urlloc>

<audio linkends="leng7 ldeutsch7 audio7">
<treeloc id="leng7"> locsrc=elink>1 2 1</treeloc>
<treeloc id="ldeutsch7" locsrc=dlink>1 2 1</treeloc>
<urlloc id="audio7">file=aquca.wav start=xxx end=yyy unit=ms</urlloc>

<audio linkends="leng8 ldeutsch8 audio8">
<treeloc id="leng8"> locsrc=elink>1 2 2</treeloc>
<treeloc id="ldeutsch8" locsrc=dlink>1 2 2</treeloc>
<urlloc id="audio8">file=aquca.wav start=xxx end=yyy unit=ms</urlloc>

....
```

FIG. 20

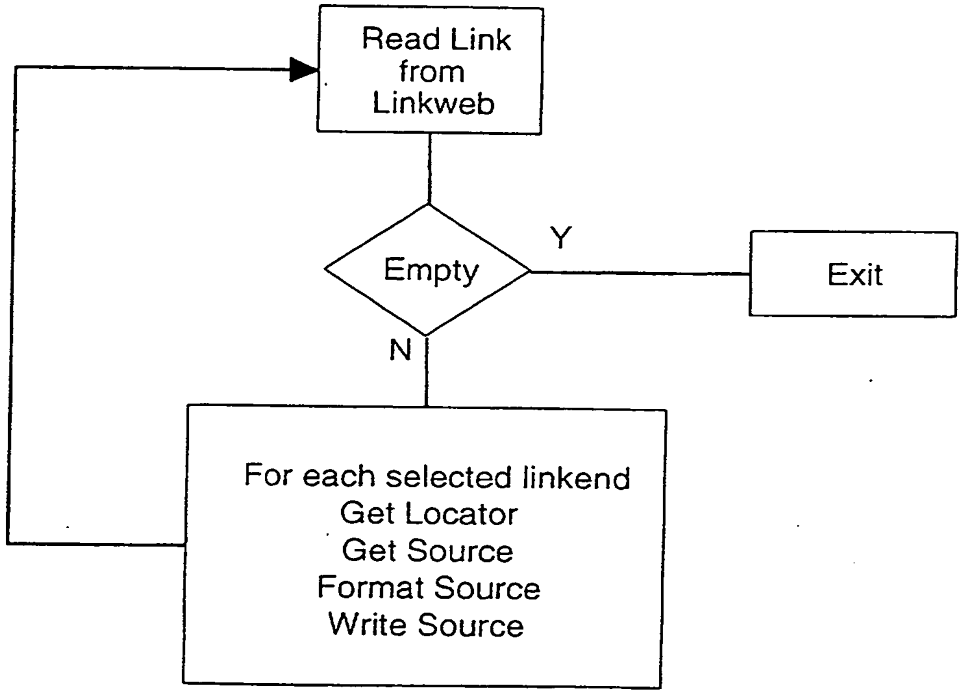
Read Link
from
Linkweb
Empty
Y
Exit
N
For each selected linkend
Get Locator
Get Source
Format Source
Write Source

FIG. 21

```
<![ CDATA [
<!DOCTYPE linkweb SYSTEM "linkweb.dtd" [
  <!ENTITY elink SYSTEM "leone.sgm" CDATA SGML>
]>

<linkweb>

<primaryLink linkends="leng1  audio1">
<treeloc id="leng1"> locsrc=elink>1</treeloc>
<urlloc id="audio1">file=aquca.wav start=200 end=3579300 unit=ms</urlloc>

<primaryLink linkends="leng2 audio2">
<treeloc id="leng2"> locsrc=elink>1 1</treeloc>
<urlloc id="audio2">file=aquca.wav start=200 end=17100 unit=ms</urlloc>

<primaryLink linkends="leng3 audio3">
<treeloc id="leng3"> locsrc=elink>1 1 1</treeloc>
<urlloc id="audio3">file=aquca.wav start=200 end=7100 unit=ms</urlloc>

<primaryLink linkends="leng4 audio4">
<treeloc id="leng4"> locsrc=elink>1 1 2</treeloc>
<urlloc id="audio4">file=aquca.wav start=7100 end=12400 unit=ms</urlloc>

<primaryLink linkends="leng5 audio5">
<treeloc id="leng5"> locsrc=elink>1 1 3</treeloc>
<urlloc id="audio5">file=aquca.wav start=12400 end=17100 unit=ms</urlloc>

<primaryLink linkends="leng6 audio6">
<treeloc id="leng6"> locsrc=elink>1 2</treeloc>
<urlloc id="audio6">file=aquca.wav start=17100 end=27900 unit=ms</urlloc>

<primaryLink linkends="leng7 audio7">
<treeloc id="leng7"> locsrc=elink>1 2 1</treeloc>
<urlloc id="audio7">file=aquca.wav start=17100 end=23500 unit=ms</urlloc>

<primaryLink linkends="leng8 audio8">
<treeloc id="leng8"> locsrc=elink>1 2 2</treeloc>
<urlloc id="audio8">file=aquca.wav start=23500 end=27900 unit=ms</urlloc>

....
]]>
```

FIG. 22

```
<![ CDATA [
<!DOCTYPE linkweb SYSTEM "linkweb.dtd" [
  <!ENTITY elink SYSTEM "leone.sgm" CDATA SGML>
  <!ENTITY dlink SYSTEM "leond.sgm" CDATA SGML>
]>

<linkweb>

<repRepLink linkends="leng1 ldeutsch1">
<treeloc id="leng1"> locsrc=elink>1</treeloc>
<treeloc id="ldeutsch1" locsrc=dlink>1</treeloc>

<repRepLink linkends="leng2 ldeutsch2">
<treeloc id="leng2"> locsrc=elink>1 1</treeloc>
<treeloc id="ldeutsch2" locsrc=dlink>1 1</treeloc>

<repRepLink linkends="leng3 ldeutsch3">
<treeloc id="leng3"> locsrc=elink>1 1 1</treeloc>
<treeloc id="ldeutsch3" locsrc=dlink>1 1 1</treeloc>

<repRepLink linkends="leng4 ldeutsch4">
<treeloc id="leng4"> locsrc=elink>1 1 2</treeloc>
<treeloc id="ldeutsch4" locsrc=dlink>1 1 2</treeloc>

<repRepLink linkends="leng5 ldeutsch5">
<treeloc id="leng5"> locsrc=elink>1 1 3</treeloc>
<treeloc id="ldeutsch5" locsrc=dlink>1 1 3</treeloc>

<repRepLink linkends="leng6 ldeutsch6">
<treeloc id="leng6"> locsrc=elink>1 2</treeloc>
<treeloc id="ldeutsch6" locsrc=dlink>1 2</treeloc>

<repRepLink linkends="leng7 ldeutsch7">
<treeloc id="leng7"> locsrc=elink>1 2 1</treeloc>
<treeloc id="ldeutsch7" locsrc=dlink>1 2 1</treeloc>

<repRepLink linkends="leng8 ldeutsch8">
<treeloc id="leng8"> locsrc=elink>1 2 2</treeloc>
<treeloc id="ldeutsch8" locsrc=dlink>1 2 2</treeloc>

....
]]>
```

FIG. 23

```
<![ CDATA [
<!DOCTYPE linkweb SYSTEM "linkweb.dtd" [
  <!ENTITY elink SYSTEM "leone.sgm" CDATA SGML>
  <!ENTITY dlink SYSTEM "leond.sgm" CDATA SGML>
]>

<linkweb>

<mLink linkends="leng1 ldeutsch1 audio1">
<treeloc id="leng1"> locsrc=elink>1</treeloc>
<treeloc id="ldeutsch1" locsrc=dlink>1</treeloc>
<urlloc id="audio1">file=aquca.wav start=200 end=3579300 unit=ms</urlloc>

<mLink linkends="leng2 ldeutsch2 audio2">
<treeloc id="leng2"> locsrc=elink>1 1</treeloc>
<treeloc id="ldeutsch2" locsrc=dlink>1 1</treeloc>
<urlloc id="audio2">file=aquca.wav start=200 end=17100 unit=ms</urlloc>

<mLink linkends="leng3 ldeutsch3 audio3">
<treeloc id="leng3"> locsrc=elink>1 1 1</treeloc>
<treeloc id="ldeutsch3" locsrc=dlink>1 1 1</treeloc>
<urlloc id="audio3">file=aquca.wav start=200 end=7100 unit=ms</urlloc>

<mLink linkends="leng4 ldeutsch4 audio4">
<treeloc id="leng4"> locsrc=elink>1 1 2</treeloc>
<treeloc id="ldeutsch4" locsrc=dlink>1 1 2</treeloc>
<urlloc id="audio4">file=aquca.wav start=7100 end=12400 unit=ms</urlloc>

<mLink linkends="leng5 ldeutsch5 audio5">
<treeloc id="leng5"> locsrc=elink>1 1 3</treeloc>
<treeloc id="ldeutsch5" locsrc=dlink>1 1 3</treeloc>
<urlloc id="audio5">file=aquca.wav start=12400 end=17100 unit=ms</urlloc>

<mLink linkends="leng6 ldeutsch6 audio6">
<treeloc id="leng6"> locsrc=elink>1 2</treeloc>
<treeloc id="ldeutsch6" locsrc=dlink>1 2</treeloc>
<urlloc id="audio6">file=aquca.wav start=17100 end=27900 unit=ms</urlloc>

<mLink linkends="leng7 ldeutsch7 audio7">
<treeloc id="leng7"> locsrc=elink>1 2 1</treeloc>
<treeloc id="ldeutsch7" locsrc=dlink>1 2 1</treeloc>
<urlloc id="audio7">file=aquca.wav start=17100 end=23500 unit=ms</urlloc>

<mLink linkends="leng8 ldeutsch8 audio8">
<treeloc id="leng8"> locsrc=elink>1 2 2</treeloc>
<treeloc id="ldeutsch8" locsrc=dlink>1 2 2</treeloc>
<urlloc id="audio8">file=aquca.wav start=23500 end=27900 unit=ms</urlloc>

....
]]>
```

FIG. 24